

REMARKS

Claims 23-55 were previously canceled without prejudice. Claims 1-22 remain in the application for consideration. In view of the follow remarks, Applicant respectfully requests that the rejections be withdrawn and the application be forwarded on to issuance.

The Rejections

Claims 1-22 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,101,508 to Wolff. For the reasons set forth below, Applicant traverses the Office's rejections.

Preliminarily, Applicant has reviewed Wolff and respectfully submits that Wolff does not pertain to a *handle administration* system as described in Applicant's specification. Importantly, Applicant can find no mention of the term "handle" in Wolff, as used in Applicant's specification—that is, as a mechanism to efficiently manage access to resources. See, e.g., Applicant's specification, page 10, lines 7-9. In addition, Applicant can find mention of the terms "reference", "dereference" and "suspend" as utilized in connection with Applicant's disclosed handle administration system, aspects of which are claimed in the claims discussed below.

The reason for this should appear apparent—Wolff does not pertain to a *handle administration system* and hence, cannot possibly anticipate the subject matter claimed in this application.

Claims 1-8

Claim 1 recites a multi-state handle administration system in which handles are capable of assuming states comprising:

- an *unassigned state* in which a handle is not assigned to a particular resource;
- an *assigned state* in which a handle is assigned to a particular resource and can be dereferenced to obtain a pointer to the resource; and
- a *suspended state* in which a handle is assigned to a particular resource but cannot be dereferenced to obtain a pointer to that resource.

In making out the rejection of this claim, the Office argues that the claim's subject matter is anticipated by Wolff. Specifically, the Office argues that the first element of the claim is anticipated in Fig. 1A and the related discussion in the specification. Wolff discusses its Fig 1A starting in column 1, line 46, which is excerpted below for the convenience of the Office.

FIGS. 1A-C show alternate embodiments of the current invention for client load rebalancing, distributed Input and Output (I/O), and resource load rebalancing. These embodiments allow more efficient, robust communication between a plurality of clients and a plurality of resources via a plurality of nodes. Resources can include, but are not limited to, computers, memory devices, imaging devices, printers, and data sets. A data set can include a database or a file system, for example. Nodes can include, but are not limited to, computers, gateways, bridges, and routers. Clients can include, but are not limited to, computers, gateways, bridges, routers, phones, and remote access devices. Clients may be coupled to nodes directly over a network. Nodes may be coupled to resources individually, or in combination, directly over a network.

In FIG. 1A, an embodiment of client load rebalancing is shown. Client load rebalancing refers to the ability of a client, enabled with processes in accordance with the current invention, to re-map a path

1 through a plurality of nodes to a resource. The re-mapping may take place
2 in response to a redirection command emanating from an overloaded node,
3 e.g. server. This capability allows the clients to optimize throughput
4 between themselves and the resources accessed by the nodes. A network,
5 which implements this embodiment of the invention, can dynamically
6 rebalance itself to optimize throughput by migrating client I/O requests
7 from overutilized pathways to underutilized pathways.

8 In FIG. 1A, a plurality of clients interface, via a plurality of nodes,
9 with a resource. A memory resource 118, nodes, e.g. utilization servers
10 104A-106A, and clients, e.g., a normal client 100A, and an aware client
11 102A are shown. Servers, nodes, and clustered file system nodes (CFNs)
12 104A-106A are connected to the storage resource through a private network
13 112. The private network can be implemented in any number of ways,
14 provided that both server 104A and server 106A can access memory
15 resource 118. The private network can include such interfaces as small
16 computer system interface (SCSI), fibre channel, and could be realized, for
17 example, with either circuit switch protocols, such as time division
18 multiplexing (TDM), or packet switch protocols such as 802x. Alternate
19 implementations of private network 112, in accordance with the current
20 invention, are set forth in each of the copending applications including
21 International Application No. PCT/US97/12843 (Attorney Docket No.
22 16598.705), filed Aug. 1, 1997, entitled "Method and Apparatus for
23 Allowing Distributed Control of Shared Resources" by inventors James J.
24 Wolff and David Lathrop at pages 9-41 and FIGS. 1-5 which are
25 incorporated herein by reference in their entirety as if fully set forth herein.

16 The servers 104A-106A are both connected, via a network 108, to
17 both the normal client 100A and the aware client 102A. The network 108
18 may include any network type, including, but not limited to, a packet
19 switch local area network (LAN) such as Ethernet, or a circuit switched
20 wide area network, such as the public switch telephone network (PSTN).

20 In operation at time $T=0$, normal client 100A is shown accessing
21 memory resource 118 via path 70 through overloaded server 104. At the
22 same time, aware client 102A is shown accessing memory resource 118,
23 via path 74, through overloaded server 104A. At time $T=1$, process 102P1,
24 implemented on aware client 102A, detects the overload condition of server
25 104A, and accesses memory resource 118 via an alternate path 76 through
server 106A. Thus, in this subsequent state, the load on server 104A is
reduced and the access by aware client 102A to memory resource 118 is
enhanced. Normal client 100A cannot initiate the processes discussed

1 above in connection with the aware client 102A and is unable to select an
2 alternate path 72 to the underutilized server 106A.

3 The detection of an overload condition on servers 104A-106A can
4 be made by processes 104PA and 106PA running on the servers.
5 Alternatively, the overload condition can be detected by the client, on the
6 basis of the round trip time for communications between aware client 102A
7 and server 104. Re-mapping of an alternate path can be intelligently
8 accomplished on the basis of an overall utilization and path table, or
9 randomly, on the basis of client queries to alternate nodes in response to an
10 overload condition. In the embodiment shown in FIG. 1A, clients
11 communicate across one network with nodes while the nodes communicate
12 across another network with resources. As will be obvious to those skilled
13 in the art, the current invention can be applied with equal advantage on a
14 single network on which clients, nodes, and resources coexist. Additionally,
15 what are shown as separate clients and nodes can alternatively be
16 implemented as a single physical unit. These and other embodiments of the
17 client load rebalancing portion of the invention will be discussed in greater
18 detail in connection with FIGS. 7A-D, 10G, and 10I. Alternatively, a
19 second resource could have a similar feature, e.g. a mirrored data set, and,
20 in this instance, a determination to redirect would redirect to the second
21 resource.

22 Nowhere can Applicant find in this excerpt, or anywhere else in Wolff, a
23 discussion of handles, let alone a handle administration system. Hence, for at least
24 this reason, claim 1 is not anticipated by Wolff.

25 The Office continues, however, and argues that the second element of this
claim is anticipated by Wolff's column 6, lines 49-55. This excerpt is reproduced
below for the convenience of the Office.

Each file system is assigned to be maintained by an administrative
server. There is only one administrative server per resource, e.g.
volume/file system, at any time. A server that is an administrative server
with respect to one file system can be a data transfer server with respect to
another file system.

1 Applicant notes that the second element of this claim effectively recites a
2 handle administration system in which handles are capable of assuming states
3 comprising "...an assigned state in which a handle is assigned to a particular
4 resource and can be dereferenced to obtain a pointer to the resource." The excerpt
5 cited by the Office does not even mention a "handle", an "assigned state" as such
6 pertains to the handle, or the notion of dereferencing to obtain a pointer to a
7 resource. As such, and for this additional reason, Wolff does not anticipate the
8 subject matter of this claim.

9 The Office then continues and argues that Wolff anticipates the third
10 element of this claim, citing to Wolff's Fig. 10B and C, column 34, lines 64-67
11 and column 35, lines 1-5. Applicant respectfully disagrees. Excerpts of Wolff,
12 starting at about column 33, line 4, are reproduced below for the convenience of
13 the Office.

14
15 FIGS. 10A-H show the processes implemented on each node in
16 order to implement load balancing, distributed I/O, and resource
rebalancing.

17 In FIG. 10A, the process associated with power up of a single server
18 in a network is illustrated (there may or may not be other servers already on
19 the network when this happens). The server being powered up is referred to
20 as the server of interest while the other servers, which are active on the
21 network, are referred to as active servers. The computer is powered up at
22 start 1000. Control is then passed to process 1002, where the volume
23 control processes and the device drivers shown in FIG. 2A are loaded.
24 Control then passes to process 1004, where the driver connected to the
25 physical volume is identified. Control then passes to a decision process
1006, where a determination is made if a clustered configuration database is
in existence on the active servers. When the determination is negative,
control passes to process 1008, where the volume control presents to an
administrator a template on which to create a clustered configuration
database table. Control is then passed to process 1010, where the new table
is stored on a device under volume control. Control then passes to process

1012. Alternatively, when the determination in decision process 1006 is positive, then control is passed directly to process 1012.

In process 1012, the clustered configuration database 120A-C (see FIGS. 5A-D) is read. Control then passes to 1013, where a variable "first time" is set to Boolean False. Control then passes to the server configuration subroutine 1014, which distributes the resources/volumes/file systems among the servers, and brings the server of interest on line. (see FIG. 10B) Control then passes to process 1016, where a logical name driver, loaded in process 1002, builds a database of available resources and paths to the resources and publishes the information in the network namespace. Control then passes to the command dispatch subroutine 1018, where commands are distributed as illustrated in FIG. 10E.

In FIG. 10B, the process associated with configuring the node and rebalancing the configuration database is shown. These processes define a load balancing function that implements these policies. The configuration is initiated at process 1030 and control is passed to decision process 1040. At decision process 1040, a determination is made if the lock 120D field is empty (see FIG. 5D). When the determination is negative, control passes to decision process 1048, where a determination is made if the node is on the network for the first time by comparing the variable "first time" to Boolean False. When the determination is negative, control passes to process 1066, where the configuration and balancing process is exited. No balancing is needed because the node is already part of the on-line. RAM-resident configuration database 120, replicated among the nodes, and because someone is already rebalancing, since the lock 120D (see FIG. 1C) is held. Thus the resources will indeed rebalance accounting for this node as well. When the determination is positive, the control passes to process 1042. In process 1042, the node determines which other server has the lock and sends that server a request to be queued as a new node on the network. Control then passes to decision process 1032, where a determination is made if the queue request was successful. When the determination is negative, the control is returned to decision process 1040. When the determination is positive, the control is passed to process 1050, where the variable first.sub.-- time is set to Boolean True. Control is then passed to process 1066, where the configuration and balance process is exited.

When the determination at decision process 1040 is positive, i.e. a lock is not present, control is passed to process 1038. At process 1038, a node identifier is written into the lock 120D field (see FIG. 5D) upon successful reservation of the sector in which the lock exists. Control then passes to process 1036, where the value for the lock field is read to confirm

1 the placement of the lock. Control is then passed to decision process 1034,
 2 where a determination is made if the value in the field corresponds to the
 3 server ID of the server being configured. When the determination is
 4 negative, i.e. when another CFN is rebalancing the servers, control is
 5 returned to decision process 1040. When the determination is positive,
 6 control is passed to decision process 1046 where a determination is made if
 7 the CFN needs a configuration database. When the determination is
 8 negative the control is passed to the balance metadata subroutine 1052 (See
 9 FIG. 10D). When the determination is positive, control is passed to process
 10 1044, where a configuration database is obtained before control is passed to
 11 the balance metadata subroutine 1052. Subroutine 1052 allows the server,
 12 having asserted master status by placing the lock on the configuration
 13 database, to rebalance the configuration database. Control is then passed to
 14 process 1054.

15 In process 1054, a queue of server rebalance requests is accessed.
 16 Control is then passed to decision process 1054, where a determination is
 17 made if any new requests for rebalancing have been made since
 18 configuration of the node has been initiated. If the determination is positive,
 19 control is passed to process 1058 which adds the requesting server to the
 20 configuration database. Control is then returned to the balance metadata
 21 subroutine 1052. If the determination at process 1056 is negative, control is
 22 passed to subroutine 1060. At subroutine 1060, the rebalanced
 23 configuration database is replicated to the other CFNs. Control is then
 24 passed to the decision process 1062, where a determination is made about
 25 the success of the replication, if the determination is negative, control is
 returned to the balance metadata subroutine 1052, because there was a node
 failure and the database needs to be rebalanced again to account for this
 fact. If the determination is positive, control is passed to process 1068.
 where the variable "first time" is set to Boolean True. Then process 1070
 sets all "needs replication" fields 440L of the resource database portion of
 the configuration database to Boolean False. Then control is passed to
 process 1064. At process 1064 the configuration database is released by
 removing the node identifier from the semaphore field and releasing the
 reservation of the sector in which the lock was located. Control then passes
 to process 1066, where the configuration and balance process is exited.

FIG. 10C illustrates the subroutine 1060 of FIG. 10B. The
 subroutine serves to insure that each node has the same copy of the cluster
 configuration database 120A-B. The subroutine is initiated at process 1080
 and control is passed to process 1082 which sets a variable "timeout" to
 Boolean False. Control is then passed to process 1083, where the nodes are
 brought to a quiet state in which all I/O is suspended. This is done by

1 sending a suspend I/O command to each node and receiving a response
2 from each. Control is then passed to process 1084, where the node sends
3 the changes the node made in the configuration database to all the other
4 nodes listed in the configuration database. It determines what to send by
5 looking at the needs replication field 440L (see FIG. 5B) for Boolean True
6 and only sends the current admin 440F-G fields to each node, thus
7 replicating the changes made in the database. Control is then passed to
8 process 1086, where the node waits for confirmation that each CFN has
9 received the changes. Control then passes to decision process 1090, where
10 the determination is made if a timeout has occurred while waiting for
11 confirmation from a particular node. When the determination is positive,
12 control is passed to process 1088, where the variable "timeout" is flagged as
13 Boolean True. Control then passes to process 1092, where the flagged node
14 is removed from the configuration database, and is assumed failed. Control
15 is then passed to decision process 1094. When the determination at decision
16 process 1090 is negative, the control is passed directly to decision process
17 1094.

18 Applicant notes that the second element of this claim effectively recites a
19 handle administration system in which handles are capable of assuming states
20 comprising "...a suspended state in which a handle is assigned to a particular
21 resource but cannot be dereferenced to obtain a pointer to that resource." The
22 excerpt cited by the Office does not even mention a "handle", a "suspended state"
23 as such pertains to the handle, or the notion of not dereferencing to obtain a
24 pointer to a resource. As such, and for this additional reason, Wolff does not
25 anticipate the subject matter of this claim.

For any of the three reasons noted above, Wolff does not anticipate the
subject matter of this claim. Accordingly, this claim is allowable.

Claims 2-8 are allowable as depending from an allowable base claim.

Claims 9-15

Claim 9 recites a handle administrator configured to manage handles that are associated with resources, the handle administrator being configured to place the handles in one of more than two possible states which affect whether a handle can be dereferenced to provide a pointer to the resource with which the handle is associated.

In making out the rejection of this claim, the Office argues that its subject matter is anticipated by Wolff in column 2, lines 49-51. This excerpt of Wolff is provided below for the convenience of the Office.

Resource rebalancing allows the network to reconfigure itself as components come on-line/off-line, as components fail, and as components fail back.

This excerpt in no way discloses or even suggests the subject matter of this claim. Specifically, nowhere does this excerpt mention a "...handle administrator being configured to place [] handles in one of more than two possible states which affect whether a handle can be dereferenced to provide a pointer to the resource with which the handle is associated."

Accordingly, this claim is not anticipated by Wolff.

Claims 10-15 are allowable as depending from an allowable base claim.

Claim 16

Claim 16 recites a handle administration system comprising [emphasis added]:

- one or more computer-readable media; and
- software code embodied on the computer-readable media which is configured implement a *handle administration system* that comprises:
 - an *unassigned state* in which a handle is not assigned to a particular resource;
 - an *assigned state* in which a handle is assigned to a particular resource and can be dereferenced to obtain a pointer to the resource; and
 - a *suspended state* in which a handle is assigned to a particular resource but cannot be dereferenced to obtain a pointer to that resource.

As noted above in connection with claim 1, Wolff does not disclose a handle administration system as recited in this claim. Accordingly, Wolff does not anticipate this claim and it is allowable.

Claim 17

Claim 17 recites a resource management system configured to manage resources comprising [emphasis added]:

- one or more resources that can be consumed by one or more agents; and
- a handle administrator configured to administer a handle system in which handles to the one or more resources are provided to the agents and can be dereferenced into pointers to the one or more resources, the handle system comprising more than two states for a handle, the states comprising:

- an *assigned state* in which a handle is associated with a resource and can be dereferenced into a pointer to that resource;
- an *unassigned state* in which the handle is not associated with a resource and cannot be dereferenced into a pointer to any resources; and
- a *suspended state* in which the handle is associated with a resource but cannot be dereferenced into a pointer to any resources.

In making out the rejection of this claim, the Office cites to the same or similar sections of Wolff as were cited to for supporting the rejection of claim 1. As noted above, Wolff does not disclose a resource management system having a handle administrator as recited in this claim. Accordingly, Wolff does not anticipate this claim and it is allowable.

Claims 18-22 are allowable as depending from an allowable base claim.

Conclusion

Applicant respectfully submits that all of the claims are in condition for allowance and, accordingly, requests a Notice of Allowability be issued forthwith. If the Office's next anticipated action is to be anything other than issuance of a Notice of Allowability, Applicant respectfully requests a telephone call for the purpose of scheduling an interview.

Respectfully Submitted,

Dated: 4/26/05

By: 

Lancee R. Sadler
Reg. No. 38,605
(509) 324-9256 ext. 226